

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. **(currently amended)** A method for improving performance of a program, comprising:
providing a call to a clone of a function from which the clone is created, wherein said call is provided regardless of whether a function body of the clone exists or not;
the function representing programming code performing a task for the program;
generating information mapping the clone to the function;
at link time for the program, if no function body of the clone is accessible by a linker, then the linker using the information mapping the clone to the function to satisfy a linker's requirement; and
at load time for the program, if no function body of the clone is accessible by a loader, then the loader, based on the information mapping the clone to the function, resolving a function reference to a body of the function; and if the body of the clone is accessible by the loader, then the loader resolving the function reference to the body of the clone.
2. **(Original)** The method of claim 1 wherein a call to the function is substituted by the call to the clone of the function.
3. **(Original)** The method of claim 2 wherein a compiler substitutes the call to the function by the call to the clone of the function.
4. **(previously presented)** The method of claim 1 wherein the mapping

information is included in an annotation section of an object of the program.

5. (Original) The method of claim 1 wherein a compiler generates the mapping information.

6. (previously presented) The method of claim 1 wherein the mapping information is stored in a loadable note section for use by the loader.

7. **(currently amended)** The method of claim 1 wherein the function body of the clone for use by the call to that clone is selected from a list of bodies based on a priority.

8. (Original) The method of claim 1 wherein the function has more than one clone in the program.

9. (Original) The method of claim 1 wherein the clone is associated with a flag identifying the clone as a function clone.

10. (Original) The method of claim 1 wherein symbol resolution of the clone is delayed to the load time for the program based on a linkage entry provided by the linker.

11. (Original) The method of claim 1 wherein a name representing the clone includes one or a combination of a condition for cloning and a name representing the function.

12. **(currently amended)** The method of claim 1 wherein the function body of the clone is included, independently of the provision of said call in said program, in a library for use by the program.

13. **(currently amended)** The method of claim 1 wherein a compiler creates the

function body of the clone based on a programming statement provided to the compiler.

14. **(currently amended)** The method of claim 1 wherein a compiler creates the function body of the clone after an analysis determining advantages and disadvantages of such creation.

15. (Original) The method of claim 1 wherein the clone is created based on one or a combination of:

a logical relationship between at least two parameters passed to the function;

an alias-relationship between at least two parameters passed to the function;

a value of at least one parameter passed to the function from; and

a number of alignment bytes of at least one parameter passed to the function.

16. (Original) The method of claim 1 wherein the clone is created based on profile data of the function.

17. **(currently amended)** A method for using a clone cloned from a function in a program, comprising:

~~using information mapping the clone to the function to satisfy~~ satisfying a linker's requirement of having a clone body for a call to the clone, regardless of whether the clone body exists or not, by using information mapping the clone to the function;

the linker's requirement being part of building the program; and

building a library that includes the body of the clone;

wherein the function represents programming code performing a task for the program and building the program and the library are independent of one another.

18. **(currently amended)** The method of claim 17, prior to building the library that includes the body of the clone, comprising building using a library that includes a body of the function but does not include the body of the clone with the program that includes the call to the clone.

19. (Original) The method of claim 17 wherein the call to the clone has replaced a call to the function.

20. (Original) The method of claim 17 wherein the clone is created based on information passed to the function.

21. **(currently amended)** A method for using a clone cloned from a section of code of a program, comprising:

substituting a call to the section of the code by a call to the clone regardless of whether a body of the clone exists or not;

at link time for the program, mapping the clone to the section of code;

at load time for the program, mapping the clone to the section of code; and

during execution of the program, if [[a]] the body of the clone is available in a library used by the program, then using that body, else if the body of the clone is not available in the library, then using the section of code from which the clone is cloned.

22. (Original) The method of claim 21 being implemented as program instructions stored in a computer-readable medium.

23. **(currently amended)** A system of programmed hardware and/or hard-wired circuitry for using a clone cloned from a function in a program, comprising:

means for mapping the clone to the function to satisfy a linking requirement of having a clone body for a call to the clone regardless of whether the clone body exists or not;

the linking requirement being part of building the program; and

means for building a library that includes the body of the clone;

wherein the function represents programming code performing a task for the program, and building the program and the library are independent of one another.

24. (Original) The system of claim 23 wherein the clone is created based on information passed to the function.

25. **(currently amended)** A computer-readable medium embodying instructions for causing, when executed by a computer, the computer to perform performing a method for improving performance of a program, the method comprising:
providing a call to a clone of a function from which the clone is created, wherein said call is provided regardless of whether a function body of the clone exists or not;
the function representing programming code performing a task for the program;
generating information mapping the clone to the function;
creating the clone based on one or a combination of
 a logical relationship between at least two parameters passed to the function;
 an alias-relationship between at least two parameters passed to the function;
 a value of at least one parameter passed to the function from; and
 a number of alignment bytes of at least one parameter passed to the function;
at link time for the program, if no function body of the clone is accessible by a linker, then the linker using the information mapping the clone to the function to satisfy a linker's requirement; and
at load time for the program, if no function body of the clone is accessible by a loader, then the loader, based on the information mapping the clone to the function, resolving a function reference to a body of the function; and if the body of the clone is accessible by the loader, then the loader resolving a function reference to the body of the clone.

26. (Cancelled)

27. **(currently amended)** The computer-readable medium of claim 25 wherein the program includes multiple calls to multiple clones of the function.